广东省高等学校“九五”规划重点教材

软件工程

龚世生 刘建宾 编著

广东高等教育出版社

广东省高等学校“九五”规划重点教材

软件工程

龚世生　刘建宾　编著

广东高等教育出版社
·广州·

内容简介

本书是软件工程学的教材，书中系统地介绍了软件工程的原理、方法、技术、标准、工具、环境以及软件工程中的管理问题。较详细地讲述了系统分析、软件设计和界面设计、编码、测试和软件维护与重用的主要概念、原则及开发方法，着重介绍结构化方法和面向对象的方法，集成化 CASE 软件环境、开发工具和文档格式，可供读者在软件开发时使用。最后，介绍了组件和分布式开发方法的新进展，以及抽象逻辑结构图程序设计技术。

本书选材适当，语言简练，条理清楚，逻辑性强，可作为大专院校计算机专业的参考书、软件工程课程的教材，也可作为计算机专业教师、研究生和有软件开发实践经验的科技人员的参考书。

图书在版编目(CIP)数据

软件工程/龚世生，刘建宾编著.—广州：广东高等教育出版社，1999.10

广东省高等学校“九五”规划重点教材

ISBN 7-5361-2414-7

Ⅰ.软… Ⅱ.①龚… ②刘… Ⅲ.软件工程 Ⅳ.TP311

中国版本图书馆 CIP 数据核字(1999)第 23598 号

广东高等教育出版社出版发行

地址：广州市广州大道北广州体育学院内 20 栋

邮政编码：510076 电话：87553782

广州市新光明印刷厂印刷

787 毫米×1 092 毫米 16 开 24 印张 555 千字

1999 年 10 月第 1 版 1999 年 10 月第 1 次印刷

印数：1～1 500 册

定价：36.00 元

序

软件工程是世界各国都重视的科学技术，被公认为今后信息产业发展的关键。这一领域正处在百花齐放、群雄竞技的时代。充分开展我们的科学研究，把软件工程的研究搞上去，是我国软件工作者的艰巨任务。计算机软件产业的竞争是人才的竞争，人才的培养不仅要在数量上、更要在质量上提高。我们应该在人才的培养上下工夫，本书是本着这个宗旨来编写的。

本书是我在编写的《软件工程引论》一书的基础上添了较多的新内容而成书的。根据在教学中的经验，对原书作了大幅度的修改，介绍了90年代的新技术，最后介绍了本书另一作者刘建宾副教授提出的一种新的工程化程序设计技术——抽象逻辑结构图程序设计技术。

本书系统地介绍了软件工程的原理、方法、技术、标准、工具、环境以及管理问题。软件工程是一门迅速发展的新学科，新的技术方法不断涌现，一本书不可能全部包容。本书在选材时征求了学生的意见，他们认为软件工程教材应以一两种方法为主，把它讲深讲透比较好，而同一开发阶段不同的方法在课堂上讲得太多不好。例如软件设计方法，如果在同一本书中，同一阶段讲它六七种不同的方法，每种方法都讲一点，效果不一定好。可以列出这些方法的参考文献，让学生自学。本书以结构化方法和面向对象两种方法为主，力争使学生能掌握好当前的主流方法。本书可作为计算机高年级本科生的教材，也可以作为研究生的参考教材。

编写本教材时，参照了国际计算机1991课程表（1990年12月ACM和IEEE/CS联合发表的“Computing Curricula 1991”），借鉴了中国计算机学会教育委员会关于软件工程的教学大纲。根据国家教育部关于计算机专业软件工程课程教学的基本要求，讲授时可按照本书的自然顺序，除有“*”外，均可对本

科生讲授。

本书为广东省高等学校“九五”规划重点教材，由陶培基副教授审校，他对本书的写法和文字提出了宝贵的意见和建议，作者对他以及对本书给予帮助的其他老师和研究生的帮助表示衷心的感谢。

最后，诚恳地欢迎广大读者对本书的缺点、错误提出批评指正。

龚世生

1998 年 5 月于汕头大学

目　录

第一章　软件工程概述

计算机技术是20世纪最重要的科学技术之一，1946年世界上制成第一台电子计算机以来，计算机技术发展十分迅速，计算机的性能平均每18个月提高一倍，成几何级数的增长，并且广泛地应用于生产，应用于社会和家庭生活。各种新技术，特别是全球互连网络Internet和多媒体技术的发展，使今天的社会进入了以计算机为核心的信息社会。信息的获取、处理、交换、决策都需要大量的优秀的计算机软件。人们对计算机软件的品种、数量、功能、质量成本和开发时间等提出了越来越高的要求。为了使计算机软件资源能够为人类共享，人们越来越重视软件和软件开发以及运行环境的标准化。软件的生产、销售和应用已成为一个庞大的行业，也可以说是极具吸引力的行业。正因为人们对软件的要求越来越高，越来越复杂，这就促使人们去克服困难，发展软件工程学。本书内容就是从理论、方法、模型、工具，环境多方面来讲解软件工程学。

第一节　软件工程学与软件危机

随着计算机的广泛应用，软件在计算机系统中占有越来越重要的地位，软件开发越来越复杂，程序人员越来越满足不了需要，软件产品质量难以满足各方面的要求，加上软件的生产率低，导致软件生产费用的上涨。以美国为例，1980年美国软件生产总费用接近400亿美元，占美国国民生产总值的2%，到1990年，软件生产费用已近2 000亿美元，占国民生产总值的8%左右，到本世纪末美国的软件产值占国民生产总值的10%以上。这说明了软件在社会生产和生活中的重要地位。

本世纪60年代末期，国际上出现软件危机，表现在软件成本急剧地上涨，成为计算机系统最大的开支项目，软件开发的周期长，进度很难控制，质量难以保证。当时，有人统计美国的大型项目都有错误，例如美国的阿波罗8号宇宙飞船，因一个计算机软件错误，造成一部分信息丢失，阿波罗14号在10天飞行中出现了18个软件错误……1971年美国的学者Hamming批评这种情况时说："我估计，即使细心地编写程序，每200条到300条指令中必定有一个错误。"由于美国缺乏软件人员，计算机公司大量招聘程序员，甚至公共汽车售票员也应招去了，因此粗制滥造的软件大量涌向市场，满足不了软件的质量要求。1968年著名的计算机科学家Dijkstra在NATO会议上说："整个的事业是建立在一个大骗局上。"当时计算机科学家称这种现象为软件危机。所谓软件危机就是指软件成本高、质量低、不能按期交付使用、可靠性差、生产效率低等状况。

为了解决这一危机，美国和西欧的一些计算机科学家，于1967年、1968年在欧洲召开了两次软件可靠性国际会议。在1968年的会议上第一次提出了"软件工程"这个词和一

些软件工程的技术。力图借助于现代工程原理来开发计算机程序及有关资料。其目的是要提高软件开发的生产率和软件产品的质量。为了开发计算机软件，许多专家和教授都在探讨软件工程的概念。叶祖尧教授指出："软件是一个不断发展和完善的概念。最初所指的软件就是计算机程序，而系统程序才称得上软件。随着时间的推移，人们对'软件'一词的认识逐渐深化，它不仅包括了所有计算机程序，而且包括了所有的过程、规则及文件资料"。根据国际标准化组织的定义，软件是"与计算机系统操作有关的程序、过程、及任何有关的文档资料。"与此同时，人们对软件的开发，也对人用手工编制程序逐渐感到不适应。在实践中，人们逐步认识到：正如不能用制造独木船的手工方式来研制航空母舰一样，也不能用个人编制小程序的方式来研制大型软件系统，于是软件工程学的研究就应运而生。从70年代初，IBM公司运用软件工程的方法研制了两个大型系统软件：一是纽约时报的情报检索系统，它使用了结构程序设计的方法，使系统很成功；另一个是空间实验室的飞行模拟系统，尽管系统庞大，用户要求又有很多变化，人力和经费有限，由于采用了工程化的技术，按时高质量地完成了系统的研制工作，软件生产率提高了一倍。到70年代末已经提出了软件工程的理论和方法及工具系统，初步形成了软件工程这门学科。

软件工程的发展可以分为四个时期，即程序设计时期，软件时期，软件工程形成时期和软件工程发展时期。关于各个时期的特点，说明如下。

1．程序设计时期(1946～1955)

这个时期的主要特点是程序设计人员个体手工编制程序，最初软件作为硬件的一种附属品，只是为了计算机能运行和作一些简单的计算及数据处理，程序还没有作为商品，工作效率很低，属手工业生产方式。

2．软件时期(又称程序系统时期 1955～1970)

这个时期的主要特点是互助组式的生产方式，主要采用由程序员小组进行编程的方法，当时社会上大量需求软件，但软件人员缺乏，而且水平也低。价格高，质量低，粗制滥造的软件大批涌向市场。有人又把这个时期称为软件危机时期或人力奇缺时期。

3．软件工程形成时期(1970～1990)

软件作为一种社会产品，批量生产，有标准化的生产过程，出现了大批软件公司，软件工厂，以软件作为计算机的中心，提出了一整套软件生产过程的基础理论、方法、工具系统。

4．软件工程发展时期(1990以后)

软件工程的思想、方法、技术在实践中得到很大的发展，大大地减少了开发费用，提高了软件的质量，面向对象技术，CASE技术，软部件组装等软件开发新的技术逐渐成熟，"软件工程"这门富有潜力的学科就这样蓬勃发展起来了。

软件工程学研究的内容是："如何应用一些科学理论和工程上的技术来指导软件的开发，从而达到较少的投资获得高质量的、可靠的软件的目的."它涉及软件生产有关的所有活动和学科，其范围相当广，包括计算机科学、管理学、经济学、心理学、可靠性理论等。本书只讨论软件开发的主要技术问题。

软件工程学研究程序设计的方法学，后来发展为结构化程序设计和模块化程序设计的方法，70年代后期开始重视软件工具，例如UNIX工具系统，根据方法论与系统的结合，

逐步形成现代化的软件开发环境。

软件工程的主要目的之一是提高软件的可靠性，软件的可靠性是指在所给条件下和规定的时间内，能完成所要求功能的性质。有时为了衡量它，用所谓软件可靠度，即软件在所给条件下和规定时间内，能完成所要求功能的概率。

显然，软件可靠性和硬件可靠性是完全不同的两个概念。对硬件而言，经过早期故障排除之后，它稳定在一定故障范围内，这个期间故障发生是随机的，通常是按 Roieon 曲线分布的。经过若干年使用之后，故障率会增大(图 1.1)。而软件本身是无磨损的，工作中不会发生损耗而产生新故障。软件的故障是在设计、制作阶段造成的。软件的故障与当时的输入系统有关，是系统状态的函数，是一种系统的现象。软件随着时间推移而故障减小，图 1.2 的曲线是在理想的前提下，即假定在一个故障修复时期内，不再出现另外的故障。图中，r 代表故障，t 代表时间。

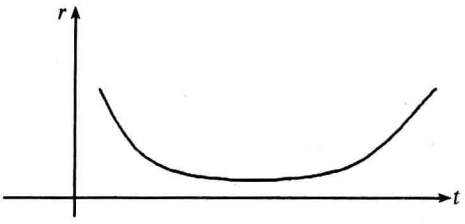

图 1.1　硬件系统故障率曲线

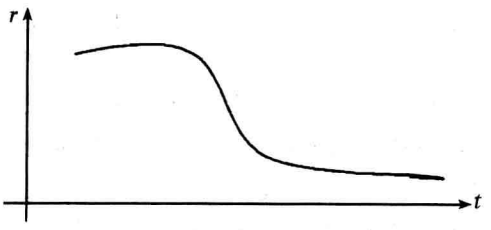

图 1.2　软件系统故障率曲线

第二节　软件工程的基本概念

软件工程就是用科学和工程的方法与原则来研制、开发和维护计算机软件的有关技术和管理方法、它应包括方法，工具和过程三个部分。下面我们分别讨论这些概念。

一、工程化方法

在解决软件危机时，公认必须采取软件开发工程化的方法。什么是软件工程化方法呢？主要有以下几个特点。

①在定义、开发和维护的每一个阶段，都采用经过验证行之有效的方法。在研制软件的初期人们以一种个体手工艺方法来开发，没有任何限制。而在软件工程时期，软件不再是个人的成果，而是集体劳动的产品。所以，软件产业的重要方法就是规范化。采用一套经前人多次验证，认为有效的方法，并按一定程序和格式书写文档，规定了明确的工作步骤。

②为使软件研制走上工程化的轨道，人们就要寻找和遵循一些标准化的规程，对开发人员给以指导和约束，使他们遵循规定的方法来理解和处理问题，这是软件开发成功的重要因素。

③在软件开发的每一个阶段，都要做复审，以保证开发出来的软件的质量。在需求分析、设计、编码等阶段都必须对初步方案进行多次的复审，尽量减少工作中的失误。

④每个阶段都要产生必要的文档。工程化必须强调文档化，即每个软件人员都必须将每一步的工作结果以一定的书面格式记录下来，存入档案。格式可以是自然语言写的文件资料，也可以是图形和图表形式。这将保证开发人员之间有效的进行交流，也有利于维护工作的顺利进行。软件开发方法应该规定描述产品的文档格式，文档中应记录哪些内容，采用哪些表格、符号等。

⑤广泛使用各种软件工具和环境。和其他工业生产相似，生产力的发展依靠生产工具的改进，软件开发必须使用工具软件，才能提高开发的速度，节省人力。一些成熟的方法，如结构程序方法和面向对象的方法等，它们的每一个阶段都要使用一定的软件工具。

⑥提供从原始产品概念到最后产品制造的一个可追溯的途径。我们要推行一套可以从原始数据和计划到运行程序的一套配置管理，既能从目标程序追溯到需求定义，又能从每个功能检查到每一个程序模块。这是软件生产过程中不可缺少的方法。

以上的方法是软件生产“高速、高产、优质”的有力保证。

任何生产规程都经过了许多年的发展历史，诸如机械工程、建筑工程已经发展成为人们可以接受的工程学。而软件工程也处在走向工程科学的途中，近 30 年来，软件工作者研制出了许多工程化方法，例如 70 年代初的编写程序的结构化程序设计方法，确实起到了提高效率，减少错误的效果。但在 70 年代中期，软件工作者认识到编写程序仅仅是软件开发的一个环节，而合理地建立系统结构比编写程序更为重要。所以研究的重点前移到设计阶段，出现了设计阶段的 SD 方法和 JACKSON 等方法，到了 70 年代后期，人们又发现事先对用户的需求进行分析更为重要，故又把重点前移到分析阶段，出现了用于分析阶段和设计的面向对象的方法，SA，SADT，ISDOS 等方法。目前用什么方法来说明用户的需求、用什么方法对软件进行设计，测试和维护，仍然是软件工作者感兴趣的课题。

二、软件的评价标准

软件开发的目的是获得高质量的软件。如何来评价软件的质量高低呢？质量与生产方式有关，从初期取决于个人业务水平的高低（即个人效率和正确性），到中期取决于程序员小组集体水平的高低，最后发展为取决于软件工程中的质量管理法、软件生产管理水平的高低。我们将从易维护性、可靠性、效率、易理解性和质量管理以及生产管理方面来全面的评价，综合为以下六条。

1. 易维护性

经过测试并已交付使用的软件，在运行中还会逐步暴露出错误，还需继续排错、修改和扩充。这种维护工作量相当大。例如 IBM 公司 OS/360 系统第一版中有 1 000 个大大小小的错误，因而维护工作是相当困难的。有人统计，若一次要修改 5 ~ 10 个语句，则修改成功的可能性是 50%，若一次修改 40 ~ 50 个语句，则修改成功可能性下降到 20%，因此软件维护是困难的。

随着软件规模的扩大和复杂性的增加，也由于人工费用所占比重的上升，软件维护问题已显得越来越重要了，易维护性已成为衡量软件质量的重要标准。

“易维护性”通常指一个程序易阅读、易发现和纠正错误、易修改扩充的程度。为了使软件容易维护，在开发早期就必须采取一定的技术，仔细分析，精心设计，并建立完整的文档。如果在早期不认真做好这些工作，就可能在后期出现许多测试和维护上的问题，由于维护和测试在总投资中占的比例很大，故防患于未然十分必要。

2. 可靠性

一般包括两个概念：正确性和健壮性。

一是正确性，其含义是指软件系统本身没有错误。有人定义为：软件符合需求定义。在预期的环境条件下，正确地完成期望的功能。或用逻辑的语言来说：“对一切满足输入谓词的程序，应该保持它终止、并在程序执行完成时，输出谓词得到满足”。

二是健壮性，其含义是指在硬件发生故障或输入数据不合理等意外环境条件下，系统仍然能继续正常的工作。一个健壮性好的软件应当具有抵御各种错误信息的能力。

正确性是一个比较容易受条件限制的概念，对一个小型的程序，我们可以要求它是完全正确的。而一个大型软件系统，因所处的环境条件千变万化，几乎是无限的。例如硬件系统部分可能出现多种类型的故障，用户也可能有意无意地输入一些不合理的数据，使系统遭到意想不到的破坏，所以我们不能期望一个大型系统在任何情况下都是正确的。因此就需要引进一个新的概念——健壮性。一个健壮的程序，在遇到意外情况时，能按某种预期的方式，作出适当的处理。例如它能意识到发生了意外，及时通知管理人员，并有效地控制事故的发展，不致于丢失重要信息；它也可能较快地从故障中恢复，避免严重的后果。

总的说来，“可靠性”包含了下述的意思：在预料的情况下，能正确的工作。在意外的情况下，也能作出适当的处理，不致造成严重损失。

一个正确的程序不一定是可靠的程序，因为它不检查输入数据是否合理，如果输入数据不合理就有可能造成严重的后果，所以这个正确的程序并不一定可靠。而一个可靠的程序却不一定是正确的。如一个计算工资的程序，它可能计算出某种很难出现的病假折扣情况，但它对输入数据及系统内部的状态要进行仔细的核查，因而在大多数情况下能够可靠的工作。所以，一般对程序的实用程度提作可靠性更为妥当。

3. 工作效率

效率是指系统能否有效地使用计算机资源，如计算时间和占用的空间。近年硬件价格下降，故追求效率并不如以往重要。又因追求效率与追求易维护性和可靠性往往相互抵触。片面的强调时间和空间，设计出来的系统可能结构复杂，难以理解和修改。追求可靠性需付出一定的时空代价。因此，工作效率已经不是最重要的因素。

4. 易理解性

软件在测试、排错和修改的时候，经常要供人阅读，所以质量好的软件必须是容易理解的，只有容易理解的程序，才能易于维护和修改。

易理解性有两种含义，一方面是指系统的内部结构清晰，对软件人员来说易于理解和阅读。另一方面也指系统人机界面友好，简明，清晰，对于用户来说易于使用。

5. 软件质量管理

为了使高质量的软件实用化，必须在全面质量管理的思想指导下，在软件生存周期各

阶段(如需求定义、设计、编码、测试、维护)中采取适当的提高质量的措施来考虑程序的完成度和整个软件系统的综合质量,可以通过软件系统是否符合使用目的和功能、是否满足要求的性能、异常时的可靠性和承受超负荷能力的程度、使用和维修是否方便和容易等方面评价。程序的好坏也许可以利用程序中残存的错误作粗略的定量评价,而软件的质量还要考虑服务的适用性、操作性 、维护性、可靠性和其他方面的因素,实际上只能是定性的评价。

应该指出,易维护性、可靠性等几个目标之间是有联系的,也是矛盾的。例如,可靠性和易维护性有矛盾,易维护性和效率也有可能有矛盾。在软件开发中,开发人员应充分考虑到各种可能的候选方案,在各种矛盾中作权衡。在一定条件下,使所期望的目标最大限度地得到满足。

6. 软件的标准化

软件生产过程中的标准化问题是一个十分重要的问题。当然最重要的是国际标准和国家标准。一个软件公司,它的各生产阶段也都有一定的标准,如需求定义的标准、设计标准……而且目前每个公司都有自己的统一标准,许多软件公司在生产的各个阶段制定出统一的标准,在各车间执行,确实提高了软件的生产率。同时也便于阅读,便于检查错误,便于程序员之间的交流,便于主程序员及程序员的交流和联合调试,同时也便于维护。这样运用标准化可以使从事软件开发的多数人之间所发生的各种矛盾减少到最低限度。

研究标准化问题,可以分为以下几个方面的内容,首先制订软件开发和文件编制的各种标准,其次提出各种标准化的实施步骤,要求统一的接口标准,还要探讨如何监督与评价各种标准。最后要强调各级各类程序员的标准化知识的教育。在软件开发的各阶段都要推行标准化方法。

三、软件开发工具

软件开发本身也可以看成是一种数据处理工作,或者叫符号处理工作。例如设计阶段的任务是将系统说明书加工成模块说明书,而编码阶段的任务是将模块说明书加工成程序。当然人们都希望这种过程由计算机自动地完成,达到"高速、优质、低价"的目标。

软件工具就是软件开发、分析、测试、维护的生产过程中使用的程序系统。例如操作系统就是软件工具,又如编译程序也是软件工具,它的加工对象是高级语言写的源程序,而加工结果是机器指令或者汇编语言程序。

例如,测试阶段的软件工具有:静态分析工具(如覆盖监视工具、测试数据产生器、排错程序、跟踪程序等等)。设计阶段和分析阶段的工具有 PSL/PSA 系统等。

以上介绍的各阶段软件工具组成了"软件工具箱",在软件开发的各个阶段,人们可以根据不同的需要,从"工具箱"中选择合适的工具来使用。这样,将提高软件生产速度并改进产品的质量。

但是,软件开发过程是复杂的,目前还不能完全用工具来代替,许多工作需要人的创造能力,软件开发的全部自动化是很困难的。计算机的工具软件只能起到辅助的作用。软件方法和软件工具有着紧密的联系。方法是主导,工具是辅助的。软件方法提出了明确的工作步骤和标准文档格式,这是设计软件工具的基础。所以,软件方法是研究工具的

先导,而工具的实现又将促进方法的发展。

在软件工程的发展过程中,经历了软件工具,软件工具箱,软件工作台,软件开发环境的发展过程。1975 年在第一届国际软件工程会议上提出了软件工具的概念。自第二届会议以来出现了各类软件支撑系统。它包括了若干个软件工具,我们把它称为软件工具箱。在第二届会议(2-1CSE)的"AN INTRODUCTION TO THE PROGRAMMERS WORK-BENCH"论文中提出了工作台的概念。所谓工作台,实际上是支撑软件开发的特别的专用计算机。而且开发的软件可以在异种机上运行,如 PDP - 11/UNIX 工作台上可以开发微机上的软件。

四、软件工程环境

软件开发的速度在很大程度上取决于所采用的支撑环境,就是在基本的硬件与宿主软件的基础上,为了开发系统软件与应用软件要使用的一组软件。一般指一些高档软件工具的有机结合,并在工程环境中合理地处理了人的作用,目的是提高产品软件的质量和劳动生产率。

我们建立软件工程环境一般要求达到以下几个目标:第一,支撑整个软件的生存期,它不但能够支撑开发与维护期的各个阶段,而且能够支持从需求分析,功能表达,设计,编码,测试到维护的各个阶段。第二,支撑大型软件项目,不仅能支撑个别程序的开发,而且能够支撑大型软件的开发与维护,支撑其中的所有程序。第三,工程环境中存在一种基本语言,最好是开发过程中项目软件的目标语言,而且软件工程环境也是用它来书写的。第四,支持软件开发和管理的一切活动,包括生产和配置管理,应使该环境能够支撑数据库中各个对象的配置。

对软件开发环境提出如下一些要求:首先它包括了一个宿主机的操作系统,以便于支撑环境本身的移植。其次要有一个关于各软件对象的数据库,用于存放所支撑的各个项目的自下而上软件生命周期中全部必要的信息,在软件工程环境中有良好的各个用户与各工具间的接口。还应该有可扩充的工具组与命令语言。

软件开发环境的概念,是从第四届国际软件工程会议开始出现的。第五届会议认真地讨论了这一概念。1982 年起发表了一批软件开发和具体环境系统。第七届会议上(1984 年)不仅发表了大量的软件工具和软件开发环境方面的论文,而且展出了近 20 个工具与环境系统,使全世界的软件界都重视这一新的突破。它标志着软件开发出现了新的局面。

目前,已经出现的软件工程环境,具有三级结构。第一是核心级,一般说来有核心工具组,数据库,通讯功能,运行时刻的支撑功能和机器无关的移植接口,它相当于一台基本语言书写的工具。第二级是基本级,它可以作环境的用户(即程序开发人员)使用的工具组,如编译程序,编辑程序,连结装配程序,作业控制语言的解释程序,配置管理程序,这些工具都是用基本语言写的,(例如 ADA 语言),并且都是由核心级支撑的。第三级是应用级,一般以一种特定的基本级为基础。但也包含一些补充工具,这些工具支撑项目成员所使用的特定方法,以便更好地生产各种应用软件。

现在,国际上出现的几个典型的软件开发环境有:EPOS、GANDAF 系统(美国卡乃基梅

隆大学)、康乃尔程序综合器、ADA 程序设计环境、APSE、UNIX 开发环境、美国防部的 STARS 计划、英国的 ALEUY、日本的日立公司的 SEWB3 等。软件工程环境还提供各种支撑工具,如需求分析工具、设计工具、测试工具等软件的开发工具。

从现有的工具或环境的水平来看,它们都属于半自动的交互式系统,即只能在一定的程度上帮助人们设计,修改,验证,检查文档和程序。例如数据处理的应用程序的生成,只需要使用超高级语言描述用户要求。系统自动显示其数据流程图、设计图,然后直接生成程序……我们将在第十章讨论 CASE 环境。

第三节　软件过程

第一节我们介绍了产生软件危机的原因,澄清了对软件工程的一些概念的模糊认识,但是软件工程是一项综合性的工程,它是一种系统工程。建立起关于软件开发和维护的正确观念,仅仅是解决软件危机的开始,全面解决软件危机还有很大的距离,软件工程应该是一种组织良好、管理严格,各类人员协同配合、团结完成的工程项目。几十年的经验证明,软件的工业化生产,必须有个工业标准,指导软件生产的全过程。软件工程是指导计算机软件开发和维护的工程学科。采用工程的概念、原理、技术和方法来开发与维护软件,把经过使用考验而证明正确的管理技术和当前能够采用的最好的技术方法结合起来,这就是实践软件工程。

软件工程实践应包括一个软件生存周期过程,软件生存周期过程是指计算机的输入转换成输出的一组相关活动,按照国际标准化组织(ISO)与国际电工委员会(IEC)共同制定的国际标准"ISO/IEC/2207 信息技术 - 软件生命周期"于 1995 年 8 月 1 日发布的文件,该标准为全球软件产业界商讨,洽谈计算机软件产品,研制和管理软件提供了一个基本的框架。因而得到各国软件界的高度重视。我国也制定了相应的标准 GB/T8566 - 1995《信息技术——软件生存周期过程》。这些标准是软件工程学中定义软件生存周期的主要依据。

软件过程又称为软件生存周期过程,就是指软件生存周期中一系列的相关活动,而每一个过程又包含了实现某些目标必须采取的活动集。这些活动的执行是有序的、可重复的、并行的和嵌套的。软件过程按性质分三类:基本过程、支持过程和组织过程。

一、基本过程

可分为下列五个过程。

1. 获取过程

定义广义的需求活动,即需要一个系统、软件产品或要求软件服务的组织所要求的活动。

2. 供应过程

定义供应软件这一方的活动,即向需方提供系统、软件产品或软件服务的组织为提供系统和软件而进行的活动。